REMARKS

Favorable reconsideration of this application is respectfully requested in view of the following remarks. Claims 1-16 are pending in the present application of which claims 1, 5, 8, 11, and 14 are independent.

Claims 1, 2, 5, 8, 9, 14 and 15 were rejected under 35 U.S.C. § 102(e) as being anticipated by Bracha et al. (6,601,114). Claims 3, 4, 6, 7, 10-13 and 16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bracha et al. in view of Levy (6,092,147). The above rejections are respectfully traversed for at least the reasons set forth below.

Applicant's Response to Examiner's "Response to Arguments"

The rejections in the Office Action are maintained from the previous non-final Office Action. The Applicant maintains the arguments that all the features of the claims 1, 2, 5, 8, 9, 14 and 15 are not taught by Bracha et al., and that all the features of claims 3, 4, 6, 7, 10-13 and 16 are not taught or suggested by Bracha et al. in view of Levy. The Applicant's remarks concerning each rejection are repeated from the Applicant's previous Response and provided below for your convenience.

The Applicant will now address the Examiner's Response to Arguments as set forth on pages 10 and 11 of the Office Action. On pages 10-11 of the Office Action, the Examiner's arguments state,

> Applicant's arguments with respect to claims 1, 5, 8 and 11 have been considered but that the arguments are not persuasive.
> **In the remarks the applicant has argued that:**
> (i) For claims 1, 5 and 8, cited reference [Bracha et al.] does not suggest the limitations (1) generating a plurality of signatures based on said instructions [within a compiled program], (2) each type signature indicating each input type constraint, and (3) each type signature indicating each output type description for a respective one of said instructions.

Examiner's response:

(i) Regarding the limitations as cited in claims 1, 5 and 8, applicant agrees that Bracha does disclose the verification process of module/class using the digital signature (applicant's amendment, page 10). Bracha verifies module-by-module using digital signature at pre-runtime in which it would be obvious to verify the instructions within the module/class. Thus, Bracha does disclose the limitations cited in the claims. Applicant only makes general allegations do not point out any errors in the rejection. Therefore, the rejection is proper and maintained herein.

The digital signatures in Bracha et al. are used to verify the identity of a source. Pre-verification constraints, *e.g.*, stored in a file, or a module itself can have an attached digital signature *to reliably identify the source* of the module or constraints. The digital signatures of Bracha et al. are not generated based on the instructions. Instead, a digital signature disclosed by Bracha et al. is generated to identify a source providing a module or file containing pre-verification constraints. Bracha et al. is silent as to how the digital signature is generated, but Bracha et al. fail to teach that the digital signature is generated based on instructions of a compiled program. In addition, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction.

The Examiner's Response states, "Bracha verifies module-by-module using digital signature at pre-runtime in which <u>it would be obvious</u> to verify the instructions within the module/class." The Examiner's statement apparently suggests that claims 1, 5 and 8 now stand rejected under an obviousness-type rejection. However, the Office Action rejects claims 1, 5 and 8 under 35 U.S.C. 102(e) as being anticipated by Bracha et al. rather than under 103. Thus, the rejection of claims 1, 5 and 8 has been changed, and the finality is improper.

In addition, even if it would be obvious to verify the instructions within the module/class, as alleged by the Examiner, Bracha et al. still fails to teach or suggest all the features of claims 1, 5, and 8. In particular, Bracha et al. fails to teach or suggest (1) generating a plurality of signatures based on said instructions [within a compiled program], (2) each type signature indicating each input type constraint, and (3) each type signature indicating each output type description for a respective one of said instructions, as described in detail below.

Furthermore, in response to the Examiner's statement that "Applicant only makes general allegations do not point out any errors in the rejection", the Applicant respectfully submits that the Applicant's Remarks section with respect to the 102 rejection sets out in detail the errors in the rejections in the Office Action. In particular, the Applicant sets out in detail which claimed features are not taught by Bracha et al. and why these features are not taught by Bracha et al. In fact, the Examiner has only made general allegations that the features of claims 1, 5 and 8 are taught by Bracha et al. With the exception of making an obviousness argument in the Examiner's response, the Examiner fails to point out how Bracha et al. teaches generating a plurality of signatures based on said instructions [within a compiled program]. The digital signatures in Bracha et al. are generated to identity a source and are not generated based on instructions in a compiled program.

Also, the Examiner fails to point out how each of the digital signatures of Bracha et al. indicate each input type constraint or indicate each output type description for a respective one of said instructions. The Examiner alleges that Bracha et al. verifies module-by-module using digital signature at pre-runtime in which it would be obvious to verify the instructions within the module/class. Verifying a source providing a module in Bracha et al. using a

digital signature does not include a digital signature indicating an input type description or indicating an output type description for a respective one of said instructions. Instead, the digital signatures of Bracha et al. identify a source rather than indicating an input type description or indicating an output type description for a respective one of said instructions. Thus, claims 1, 2, 5, 8, 9, 14 and 15 are believed to be allowable.

The Examiner also provided arguments concerning the 103 rejection. In particular, page 11 of the Office Action states,

> **In the remarks, the applicant has argued that:**
> (ii) For claim 11 combination of Bracha and Levy does not teach or suggest the limitation translating said instructions into a plurality of type signature and composing said type signatures into a single composed type signature as claimed.
> **Examiner's response:**
> (ii) Regarding the limitations as cited in claim 11. It is noted that the rejection clearly points out where the combination of Bracha and Levy teach the claimed features and why it would have been obvious to combine their teachings. Applicant only makes general allegations do not point out any errors in the rejection.

Neither Bracha et al. nor Levy teach or suggest composing a plurality of signatures into a single composed signature. Levy was combined with Bracha et al. to allegedly teach this feature. However, Levy also fails to teach or suggest this feature. The rejection cites column 7, lines 25-34 to teach this feature. In this passage, Levy discloses generating a proof of authenticity for bytecode using any type of cryptographic computation, such as hashing. However, Levy fails to teach or suggest composing a plurality of type signatures into a single composed type signature, wherein the plurality of type signatures are translated from instructions. The hashing and cryptographic computation of Levy do not include composing a plurality of type signatures into a single composed type signature. If this rejection is maintained, the Examiner must indicate how the hashing and cryptographic computation of

Levy can be considered to include a plurality of type signatures composed into a single type signature.

Finality Must Be Withdrawn

The finality of the Office Action must be withdrawn because the Examiner has apparently changed the 102 rejection into a 103 in the Examiner's Response to Arguments. The Examiner's Response states, "Bracha verifies module-by-module using digital signature at pre-runtime in which it would be obvious to verify the instructions within the module/class." The Examiner's statement apparently suggests that claims 1, 5 and 8 now stand rejected under an obviousness-type rejection. However, the Office Action rejects claims 1, 5 and 8 under 35 U.S.C. 102(e) as being anticipated by Bracha et al. rather than under 103. Thus, the rejection of claims 1, 5 and 8 has been changed, and the finality is improper because the Applicant must be given the opportunity to respond to the new rejection. In addition, finality must be withdrawn because the cited references fail to teach or suggest all the features of the claims as described herein.

Claim Rejection Under 35 U.S.C. §102

The test for determining if a reference anticipates a claim, for purposes of a rejection under 35 U.S.C. § 102, is whether the reference discloses all the elements of the claimed combination, or the mechanical equivalents thereof functioning in substantially the same way to produce substantially the same results. As noted by the Court of Appeals for the Federal Circuit in *Lindemann Maschinenfabrick GmbH v. American Hoist and Derrick Co.*, 221

USPQ 481, 485 (Fed. Cir. 1984), in evaluating the sufficiency of an anticipation rejection under 35 U.S.C. § 102, the Court stated:

> Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim.

Therefore, if the cited reference does not disclose each and every element of the claimed invention, then the cited reference fails to anticipate the claimed invention and, thus, the claimed invention is distinguishable over the cited reference.

Claims 1, 2, 5, 8, 9, 14 and 15 were rejected under 35 U.S.C. § 102(e) as being anticipated by Bracha et al.

Claim 1 recites,

> a code verifier configured to analyze instructions of said program and to generate a plurality of type signatures based on said instructions, each of said type signatures indicating each input type constraint and each output type description for a respective one of said instructions, wherein said code verifier is configured to detect a type error by analyzing said type signatures.

Bracha et al. fails to teach or suggest (1) generating a plurality of signatures based on said instructions, (2) each type signature indicating each input type constraint, and (3) each type signature indicating each output type description for a respective one of said instructions.

Bracha et al. discloses lazy linking with module-by-module verification. Bracha et al. discloses in the description of related art that linking is the process of taking a class at run time and combining it with a virtual machine (VM) so the module can be executed. Lazy linking loads a class, *e.g.*, an instance of a module, only at the time the class is first necessary to execute an instruction of the class. Verification is performed every time the class is linked. Verification ensures that illegal operations are not attempted by a VM that can lead to

meaningless results. See column 4, lines 28-53. However, in certain instances, when a class is referenced by another class, the class may be verified even if it is not used. Also, verification is performed at run time, so a class that has been previously verified is verified again each time the class is loaded. See column 5, lines 17-28. Thus, Bracha et al. discloses a pre-runtime, class-by-class, verification process. In Bracha's pre-run time, verification process of a class, any intermodule information referencing other modules, such as subtyping relationships between classes, is assumed so that instructions in the class being verified are valid. However, the assumed information places a constraint on the referenced module that must be remembered by the VM. The pre-verification constraints are written to a file for checking later at run time. See column 16, line 45-column 17, line 28.

As cited in the rejection and further disclosed by Bracha et al., as a further option, the pre-verification constraints, *e.g.*, stored in a file, or the module itself can have an attached digital signature <u>to reliably identify the source</u> of the module or constraints. See column 17, lines 30-35. The digital signature of Bracha et al. is used to identify the source of the file containing the constraints or the module itself. For example, if the digital signature does not match the digital signature of a trusted source or entity, then the constraints or module may be considered unsafe to use. Thus, the digital signatures of Bracha et al. are not generated based on the instructions. Instead, a digital signature of Bracha et al. is generated based on the identity of a source providing a module or file containing pre-verification constraints. Furthermore, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction. Instead, the digital signatures only identify sources rather than input type constraints or output type descriptions

of instructions. Accordingly, Bracha et al. fails to teach all the features of claim 1, and claims 1-4 are believed to be allowable.

Independent claim 5 recites, *inter alia*,

> a code verifier configured to analyze a code block of said program and to translate instructions within said code block into a plurality of type signatures, said code verifier further configured to compose said type signatures into a single composed type signature and to detect a type error by analyzing said type signatures.

Bracha et al. fails to teach a code verifier configured to translate instructions within a code block into a plurality of signatures. As stated above, the digital signature disclosed in Bracha et al. identifies the source providing the module or file containing pre-verification constraints. Bracha et al. fails to teach translating instructions into type signatures. Bracha et al. also fails to teach composing a plurality of type signatures into a single type signature. In fact, the rejection of claim 11 states that Bracha et al. does not explicitly disclose composing said type signatures into a single composed signature. The rejection of claim 11, then combines Levy with Bracha et al. to allegedly teach this feature. Thus, the Examiner agrees that Bracha et al. fails to teach composing a plurality of type signatures into a single type signature. Accordingly, claims 5-7 are believed to be allowable.

Claim 8 recites features similar to claim 1, including,

> generating a plurality of type signatures based on instructions within said program, each of said type signatures indicating each input type constraint and each output type description for a respective one of said instructions.

The digital signature of Bracha et al. is used to identify the source of the file containing the constraints or the module itself. The digital signatures of Bracha et al. are not generated based on the instructions. Instead, a digital signature of Bracha et al. is generated

based on the identity of a source providing the module or file containing the pre-verification constraints. Furthermore, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction. Instead, the digital signatures only identify sources rather than input type constraints or output type descriptions of instructions. Accordingly, Bracha et al. fails to teach all the features of independent claim 8, and claims 8-10 are believed to be allowable.

Claim 14 recites features similar to claim 1, including,

> means for generating a plurality of type signatures based on instructions within a compiled program, each of said type signatures indicating each input type constraint and each output type description for a respective one of said instructions.

Bracha et al. fails to teach a code verifier for generating a plurality of type signatures based on instructions within a compiled program. As stated above, the digital signature disclosed in Bracha et al. identifies the source providing the module or file containing pre-verification constraints. The digital signatures of Bracha et al. are not generated based on the instructions. Instead, a digital signature of Bracha et al. is generated based on the identity of a source providing the module or file containing the pre-verification constraints. Furthermore, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction. Instead, the digital signatures only identify sources rather than input type constraints or output type descriptions of instructions. Accordingly, Bracha et al. fails to teach all the features of independent claim 14, and claims 14-15 are believed to be allowable.

<u>Claim Rejection Under 35 U.S.C. §103</u>

The test for determining if a claim is rendered obvious by one or more references for purposes of a rejection under 35 U.S.C. § 103 is set forth in MPEP § 706.02(j):

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Therefore, if the above-identified criteria are not met, then the cited reference(s) fails to render obvious the claimed invention and, thus, the claimed invention is distinguishable over the cited reference(s).

Claims 3, 4, 6, 7, 10-13 and 16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bracha et al. in view of Levy.

Independent claim 11 recites, translating said instructions into a plurality of type signatures and composing said type signatures into a single composed type signature. These steps are not taught or suggested by either of Bracha et al. and Levy. As stated above, the digital signature of Bracha et al. is used to identify the source of the file containing the constraints or the module itself. The digital signatures of Bracha et al. are not generated based on the instructions, and instructions are not translated to generate a plurality of signatures. Furthermore, neither Bracha et al. nor Levy teach or suggest composing a plurality of signatures into a single composed signature. Levy was combined with Bracha et al. to allegedly teach this feature. However, Levy also fails to teach or suggest this feature. The rejection cites column 7, lines 25-34 to teach this feature. In this passage, Levy discloses

generating a proof of authenticity for bytecode using any type of cryptographic computation, such as hashing. However, Levy fails to teach or suggest composing a plurality of type signatures into a single composed type signature, wherein the plurality of type signatures are translated from instructions. Thus, claims 11-13 are believed to be allowable.

Dependent claims 3, 4, 6, 7, 10 and 16 are believed to be allowable for at least the reasons stated above that their respective independent claims are believed to be allowable. Furthermore, Bracha et al. in view of Levy fails to teach or suggest many of the features of these claims, such as translating code blocks into type signatures, composing code type signatures for each code block into a single type signature, making a determination as to whether an input type constraint of a first type signature is acceptable to an output type description of a second type signature, and a type signature including a description indicative of a type of an input consumed by an instruction when the instruction is executed and another of type signature including a description indicative of a type of an output produced by another instruction when the other instruction is executed.

based on the identity of a source providing the module or file containing the pre-verification constraints. Furthermore, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction. Instead, the digital signatures only identify sources rather than input type constraints or output type descriptions of instructions. Accordingly, Bracha et al. fails to teach all the features of independent claim 8, and claims 8-10 are believed to be allowable.

Claim 14 recites features similar to claim 1, including,

> means for generating a plurality of type signatures based on instructions within a compiled program, each of said type signatures indicating each input type constraint and each output type description for a respective one of said instructions.

Bracha et al. fails to teach a code verifier for generating a plurality of type signatures based on instructions within a compiled program. As stated above, the digital signature disclosed in Bracha et al. identifies the source providing the module or file containing pre-verification constraints. The digital signatures of Bracha et al. are not generated based on the instructions. Instead, a digital signature of Bracha et al. is generated based on the identity of a source providing the module or file containing the pre-verification constraints. Furthermore, the digital signatures of Bracha et al. do not indicate each input type constraint and each output type description for a respective one of each instruction. Instead, the digital signatures only identify sources rather than input type constraints or output type descriptions of instructions. Accordingly, Bracha et al. fails to teach all the features of independent claim 14, and claims 14-15 are believed to be allowable.

Claim Rejection Under 35 U.S.C. §103

The test for determining if a claim is rendered obvious by one or more references for purposes of a rejection under 35 U.S.C. § 103 is set forth in MPEP § 706.02(j):

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Therefore, if the above-identified criteria are not met, then the cited reference(s) fails to render obvious the claimed invention and, thus, the claimed invention is distinguishable over the cited reference(s).

Claims 3, 4, 6, 7, 10-13 and 16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bracha et al. in view of Levy.

Independent claim 11 recites, translating said instructions into a plurality of type signatures and composing said type signatures into a single composed type signature. These steps are not taught or suggested by either of Bracha et al. and Levy. As stated above, the digital signature of Bracha et al. is used to identify the source of the file containing the constraints or the module itself. The digital signatures of Bracha et al. are not generated based on the instructions, and instructions are not translated to generate a plurality of signatures. Furthermore, neither Bracha et al. nor Levy teach or suggest composing a plurality of signatures into a single composed signature. Levy was combined with Bracha et al. to allegedly teach this feature. However, Levy also fails to teach or suggest this feature.

The rejection cites column 7, lines 25-34 to teach this feature. In this passage, Levy discloses generating a proof of authenticity for bytecode using any type of cryptographic computation, such as hashing. However, Levy fails to teach or suggest composing a plurality of type signatures into a single composed type signature, wherein the plurality of type signatures are translated from instructions. Thus, claims 11-13 are believed to be allowable.

Dependent claims 3, 4, 6, 7, 10 and 16 are believed to be allowable for at least the reasons stated above that their respective independent claims are believed to be allowable. Furthermore, Bracha et al. in view of Levy fails to teach or suggest many of the features of these claims, such as translating code blocks into type signatures, composing code type signatures for each code block into a single type signature, making a determination as to whether an input type constraint of a first type signature is acceptable to an output type description of a second type signature, and a type signature including a description indicative of a type of an input consumed by an instruction when the instruction is executed and another of type signature including a description indicative of a type of an output produced by another instruction when the other instruction is executed.

Conclusion

In light of the foregoing, withdrawal of the rejections of record and allowance of this application are earnestly solicited.

Should the Examiner believe that a telephone conference with the undersigned would assist in resolving any issues pertaining to the allowability of the above-identified application, please contact the undersigned at the telephone number listed below. Please grant any required extensions of time and charge any fees due in connection with this request to deposit account no. 08-2025.

Respectfully submitted,

Christopher J. DOLLIN

Dated: February 22, 2005

By [signature]

Ashok K. Mannava
Registration No.: 45,301

MANNAVA & KANG, P.C.
8221 Old Courthouse Road
Suite 104
Vienna, VA 22182
(703) 652-3822
(703) 880-5270 (facsimile)